WHAT IS CLAIMED IS:

1. An architecture for integrating data between a plurality of software applications in a factory environment comprising:

a factory system comprising:

a domain object superclass;

a plurality of first-level subclasses of the domain object superclass, an instantiation of one of the plurality of first-level subclasses corresponding to a domain object, the domain object representing an item in a factory; and

a service, the service providing an operation related to the domain object, the service comprising at least one component, each of the at least one component being operable to perform the operation related to the domain object;

and

a domain application comprising:

an implementation of one component of the at least one component of the service of the factory system to perform the operation related to the domain object.

2. The architecture of claim 1, wherein the domain application is one of a group consisting of the following:

a legacy system; and

an integrated application.

3. The architecture of claim 1 wherein

the domain application corresponds to an integrated application comprising:

a second-level subclass of one first-level subclass of the plurality of first-level subclasses of the domain object superclass of the factory system, an instantiation of the second-level subclass corresponding to an application-specific domain object;

and

the implementation of the one component corresponds to a method of the application-specific domain object, wherein the method is operable to perform the operation on the application-specific domain object, wherein the performing the operation on the application-specific domain object enables the domain object to communicate as if the operation were performed on the domain object.

4. The architecture of claim 1 wherein
the domain application corresponds to a legacy system comprising:
a data structure corresponding to the domain object;
and
the implementation of the one component corresponds to an interface to the legacy system,
wherein the legacy system is operable to perform the operation on the data structure.

5. The architecture of claim 1, wherein
the operation comprises a plurality of operations.

6. The architecture of claim 1, wherein
the service comprises a plurality of services.

7. The architecture of claim 1, wherein
each component of the at least one component of the service has a corresponding domain application providing an implementation of the component.

8. The architecture of claim 1, wherein
the service includes instructions for selecting a component of the at least one component to perform the operation, the selecting providing a selected component; and
the selected component includes instructions to perform the operation.

9. The architecture of claim 1, wherein
a component of the at least one component is an interface to the domain application.

10. The architecture of claim 9, wherein
a requesting component of the at least one component includes instructions to use the interface to request the domain application to provide data to a receiving component of the at least one component; and
the receiving component includes instructions to receive the data from the domain application via the interface.

11. The architecture of claim 10 wherein
the receiving component and the requesting component are the same.

12. The architecture of claim 10 wherein
the receiving component further includes instructions to perform the operation on the domain object.

13. The architecture of claim 1 further comprising:
a system manager for managing hardware and software in the factory.

14. A factory system for integrating data between a plurality of software applications in a factory environment, one of the plurality of software applications corresponding to a domain application, the factory system comprising:
a domain object superclass;
a plurality of first-level subclasses of the domain object superclass, an instantiation of one of the first-level subclasses of the plurality of first-level subclasses corresponding to a domain object, the domain object representing an item in a factory; and
a service, the service providing an operation related to the domain object, the service comprising at least one component, each of the at least one component corresponding to operable to perform the operation related to the domain object;
and wherein
the domain application includes:
an implementation of one component of the at least one component of the service of the factory system to perform the operation related to the domain object.

15. A domain application for integrating data between a plurality of software applications in a factory environment, one of the plurality of software applications corresponding to a factory system, the factory system including: a domain object superclass; a plurality of first-level subclasses of the domain object superclass, an instantiation of one of the first-level subclasses corresponding to a domain object, the domain object representing an item in a factory; and a service, the service providing an operation related to the domain object using a component, the service comprising at least one component, each of the at least one component being operable to perform the operation related to the domain object; the domain application comprising:
an implementation of one component of the at least one component of the service of the factory system, wherein the component is operable to perform the operation related to

the domain object.

16. A method for integrating data between a plurality of software applications in a factory environment comprising:
providing a domain object superclass in a first software application, the first software application corresponding to a factory system;
providing a plurality of first-level subclasses of the domain object superclass;
instantiating one subclass of the plurality of first-level subclasses to create a domain object, the domain object representing an item in a factory; and
providing a service that provides an operation related to the domain object, the service comprising at least one component, each of the at least one component being operable to perform the operation;
performing the operation related to the domain object using an implementation of one component of the at least one component of the service, the implementation being provided by a second software application, the second software application corresponding to a domain application.

17. The method of claim 16 further comprising:
providing a second-level subclass of one first-level subclass of the plurality of first-level subclasses in the domain application, the domain application being an integrated application;
instantiating the second-level subclass to provide an application-specific domain object;
implementing one component of the at least one component as a method of the application-specific domain object;
performing the operation on the application-specific domain object using the method;
and wherein
the performing the operation enables the domain object to communicate as if the operation were performed on the domain object.

18. The method of claim 16 further comprising:
providing a data structure corresponding to the domain object in the domain application, the domain application being legacy system;
implementing one component of the at least one component to serve as an interface to the legacy system;

requesting the legacy system to perform the operation via the interface; and
performing the operation on the data structure.

19. The method of claim 16 further comprising:
requesting the service to perform the operation;
selecting a selected component of the at least one component to perform the operation; and
performing the operation using the selected component.

20. The method of claim 16 further comprising:
requesting the domain application to provide data to a receiving component of the at least one component; and
receiving the data from the domain application by the receiving component.

21. A computer program product for integrating data between a plurality of software applications in a factory environment comprising:
instructions for providing a domain object superclass in a first software application, the first software application corresponding to a factory system;
instructions for providing a plurality of first-level subclasses of the domain object superclass;
instructions for instantiating one subclass of the plurality of first-level subclasses to create a domain object, the domain object representing an item in a factory;
instructions for providing a service that provides an operation related to the domain object, the service comprising at least one component, each of the at least one component being operable to perform the operation; and
instructions for performing the operation related to the domain object using an implementation of one component of the at least one component of the service, the implementation being provided by a second software application, the second software application corresponding to a domain application;
and
a computer-readable medium for storing the instructions for providing the domain object superclass, the instructions for providing the plurality of first-level subclasses, the instructions for instantiating, the instructions for providing the service, and the instructions for performing the operation.

22. The computer program product of claim 21 further comprising:

instructions for providing a second-level subclass of one first-level subclass of the plurality of first-level subclasses in the domain application, the domain application being an integrated application;

instructions for instantiating the second-level subclass to provide an application-specific domain object;

instructions for implementing one component of the at least one component as a method of the application-specific domain object;

instructions for performing the operation on the application-specific domain object using the method, wherein the performing the operation enables the domain object to communicate as if the operation were performed on the domain object;

and wherein

the computer-readable medium further stores the instructions for providing the second-level subclass, the instructions for instantiation the second-level subclass; the instructions for implementing, and the instructions for performing the operation on the application-specific domain object.

23. The computer program product of claim 21 further comprising:

instructions for providing a data structure corresponding to the domain object in the domain application, the domain application being legacy system;

instructions for implementing one component of the at least one component to serve as an interface to the legacy system;

instructions for requesting the legacy system to perform the operation via the interface; and

instructions for performing the operation on the data structure;

and wherein

the computer-readable medium further stores the instructions for providing the data structure, the instructions for implementing the component to serve as the interface, the instructions for requesting, and the instructions for performing the operation on the data structure.

24. The computer program product of claim 21 further comprising:

instructions for requesting the service to perform the operation;

instructions for selecting a selected component of the at least one component to perform the operation; and

instructions for performing the operation using the selected component;

and wherein
the computer-readable medium further stores the instructions for requesting, the instructions for selecting a selected component, and the instructions for performing the operation using the selected component.

25. The computer program product of claim 21 further comprising:
instructions for requesting the domain application to provide data to a receiving component of the at least one component; and
instructions for receiving the data from the domain application by the receiving component; and wherein
the computer-readable medium further stores the instructions for requesting and the instructions for receiving.